Introduction to Tensorflow

CS 510
Lecture #23
April 6th, 2018
Step Up a Level …

Open Source ML Tools Usage

Graph from altexsoft.

Think Locally

and historically

Cameron: High Level Language
Compilation for Reconfigurable Systems

Jeff Hammes, Bob Rinker, Wim Bo"hm, Walid Najjar, Bruce Draper, Ross Beveridge

Department of Computer Science, Colorado State University

1999

Figure 4. DDCF graph for Prewitt program after loop unrolling and array value propagation.
Tensorflow Learning Resources

• First if course, tensorflow.org …
• Then …

TensorFlow

The web contains many helpful tutorials on tensorflow. I have only begun to scratch the surface. That caveat offered, I found these helpful.

• TensorFlow 101 (Really Awesome Intro Into TensorFlow). This is a relatively long but I thought really excellent walk through to set the context for TensorFlow.
• TensorFlow Tutorial - How to use TensorFlow to Build a Neural Network. This is relatively shorter - just under 8 minutes - introduction with an emphasis on image recognition.
• Dan Aloni's blog post on Back Propagation with TensorFlow. We will use this tutorial to begin digging into the basics of using Tensorflow.
Installing with native pip

We have uploaded the TensorFlow binaries to PyPI. Therefore, you can install TensorFlow through pip.

The REQUIRED_PACKAGES section of setup.py lists the packages that pip will install or upgrade.

Prerequisite: Python

In order to install TensorFlow, your system must contain one of the following Python versions:

- Python 2.7
- Python 3.3+

If your system does not already have one of the preceding Python versions, install it before proceeding.
Nitty Gritty – Installing 2

Installing with Virtualenv

Take the following steps to install TensorFlow with Virtualenv:

1. Start a terminal (a shell). You’ll perform all subsequent steps in this shell.
2. Install pip and Virtualenv by issuing the following commands:

   $ sudo easy_install pip
   $ pip install --upgrade virtualenv

3. Create a Virtualenv environment by issuing a command of one of the following formats:

   $ virtualenv --system-site-packages targetDirectory # for Python
   $ virtualenv --system-site-packages -p python3 targetDirectory

where targetDirectory identifies the top of the Virtualenv tree. Our instructions...
Nitty Gritty – Installing 3

Ended in Success

Recommended by Asa Ben-Hur!
Using Conda

• First, install
• Next, update your unix path

[Hinton:tf/tutorials/Aloni] ross% which conda
/Users/ross/anaconda2/bin/conda

• Now review basic commands
  – update
  – install
  – list
More Basics - iPython

[Hinton:tf/tutorials/aloni] ross% which ipython
/Users/ross/anaconda2/bin/ipython

Python 2.7.14 |Anaconda custom (64-bit)| (default, Dec 7 2017, 11:07:58)
Type "copyright", "credits" or "license" for more information.

IPython 5.4.1 -- An enhanced Interactive Python.
?
-> Introduction and overview of IPython's features.
%quickref -> Quick reference.
help -> Python's own help system.
object? -> Details about 'object', use 'object??' for extra details.

In [1]: import tensorflow as tf

In [2]: tf.__version__
Out[2]: '1.1.0'

In [3]:
To Start - Back Propagation

Dan Aloni's blog

Back propagation with TensorFlow

(Updated for TensorFlow 1.0, at March 6th, 2017)

When I first read about neural network in Michael Nielsen's Neural Networks and Deep Learning, I was excited to find a good source that explains the material along with actual code. However there was a rather steep jump in the part that describes the basic math and the part that goes about implementing it, and it was especially apparant in the numpy-based code that implements backward propagation.

So, in order to explain it better to myself, and learn about TensorFlow in the process, I took it upon myself to implement the first network in the book using TensorFlow by two means. First, manually defining the back propagation step, and the second - letting TensorFlow do the hard work using automatic differentiation.
Time to Start Playing ...

```python
# This code is from a tutorial put together by Dan Aloni
# http://blog.aloni.org/posts/backprop-with-tensorflow

import tensorflow as tf

from tensorflow.examples.tutorials.mnist import input_data

mnist = input_data.read_data_sets("MNIST_data/", one_hot=True)

a_0 = tf.placeholder(tf.float32, [None, 784])
y = tf.placeholder(tf.float32, [None, 10])

middle = 30
w_1 = tf.Variable(tf.truncated_normal([784, middle]))
b_1 = tf.Variable(tf.truncated_normal([1, middle]))
w_2 = tf.Variable(tf.truncated_normal([middle, 10]))
b_2 = tf.Variable(tf.truncated_normal([1, 10]))

def sigma(x):
    return tf.div(tf.constant(1.0),
                  tf.add(x, b_1, w_1))`